Latin squares in experimental design

Although a Latin square is a simple object to a mathematician, it is multi-faceted to an experimental designer. The same Latin square can be used in many different situations. Remember that an experimental design consists in the allocation of treatments to experimental units; both the set of experimental units and the set of treatments may themselves be structured in some way.

Here is a Latin square of order 5. We will consider a number of different experiments where it could provide the basic structure for the design. The discussion is due to R. A. Bailey: see for example [1].

A	B	C	D	E
B	A	D	E	C
C	E	A	B	D
D	C	E	A	B
E	D	B	C	A

First example

Suppose that we want to test five drugs A, B, C, D, E for their effect in alleviating the symptoms of a chronic disease. Five patients are available for a trial, and each will be available for five weeks. Testing a single drug requires a week. Thus an experimental unit is a 'patient-week'.

The structure of the experimental units is a rectangular grid (which happens to be square in this case); there is no structure on the set of treatments.

We can use the Latin square to allocate treatments. If the rows of the square represent patients and the columns are weeks, then for example the second patient, in the third week of the trial, will be given drug D. Now each patient receives all five drugs, and in each week all five drugs are tested.

Second example

The second example is very similar to the first. Suppose that we are testing five varieties of pesticides on a square orchard with 25 apple trees. There may be differences between rows, and differences between columns, but we assume that rows and columns have the same status (e.g. the orchard is not on a hillside). Each plot occurs in one row and one column, just as each experimental unit in

the preceding example occurs in one patient and one week. But in a sense rows are 'equivalent' to columns; patients are not equivalent to weeks in the same way. The structure on the plots is a *Hamming scheme* in this case.

Again the Latin square gives an allocation of varieties to plots in the same way as before.

Third example

We have to test five different cake recipes and five baking temperatures. We can perform experiments on five days, and five timeslots are available each day. The 25 experimental units are divided into five blocks of five, while the treatments have a 5×5 factorial structure.

Again, we can assign treatments to units using the Latin square. Let us number the cells of the square from 1 to 25 as follows.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

We assume that the rows of the table represent days, so that units $1,\ldots,5$ are the timeslots on the first day. Now we use the five columns of the square to assign the five temperatures, and the letters to assign the recipes A,B,C,D,E.

Fourth example

Suppose that we have three factors each with five levels. Continuing the preceding example, suppose that there are also five different mixing times to be tested. Suppose also that we have 25 experimental units (timeslots) available, this time with no structuring on the units. There are 125 treatments, and we can only test one-fifth of them (that is, a one-fifth fractional factorial). The Latin square provides a design in which each combination of recipe and temperature, of recipe and mixing time, and of temperature and mixing time, is tested.

Label the rows of the square by temperatures; the columns by mixing times; and let the letters $A,\ldots,E$ stand for recipes as before. Each of the 25 cells of the square represents the treatment to be applied to one experimental unit.

Fifth example

The last example is a bit different; we use the Latin square as an incidence structure rather than an array.

We have 25 different cheeses to be tasted. Fifteen tasters are available; each taster can only taste and compare five of the cheeses. Thus, not all pairs of cheeses can be directly compared by the same taster; we require that any pair should be compared by at most one taster. Also, though this is not a formal requirement, it would be nice if two cheeses which are not compared with one another are both compared with a constant number of other cheeses.

Suppose further that only five tasting booths are available, so that three tasting sessions must be run.

We number the cheeses from 1 to 25, and associate each cheese with a cell of the square array according to the earlier figure. In the first tasting session, we assign to the five tasters the cheeses in the five rows of the array: that is, the first taster gets cheeses 1, 2, 3, 4 and 5; the second gets 6, 7, 8, 9, 10; and so on. In the second session, we use the columns, so that the first taster gets 1, 6, 11, 16, 21 and so on. In the final session we use the letters of the Latin square, so that the first taster gets 1, 7, 13, 19, 25 and so on.

Note that the sets of cheeses assigned to the fifteen tasters are the blocks of the net associated to the Latin square, while the three tasting sessions correspond to the parallel classes in the resolution of the net. Properties of nets ensure that our optional condition is indeed satisfied.

References

[1] R. A. Bailey, Orthogonal partitions for designed experiments, *Designs, Codes and Cryptography* **8** (1996), 45–77.

R. A. Bailey, Peter J. Cameron
May 30, 2003